Informatics 41 • Fall 2011 • David G. Kay • UC Irvine • Your name ______________________________

THIRD QUIZ

You have 15 minutes from the start of class to complete this quiz. Read the problems with care; work with deliberate speed. Don't give us more than we ask for. The usual instructions apply. Good luck!

Problem 1 (4 points)

Evaluate each of the following expressions (including the two definitions). (That is, what does DrRacket show in the interactions window when you enter the expression(s) in the definitions window and click Run?) The function `number?` returns true if its argument is a number; the functions `even?` and `odd?` return true if their argument is an even number or an odd number, respectively.

```
(define THIS-YEAR 2011)
(define LAST-YEAR 2010)
```

(a) `(or (odd? THIS-YEAR) (odd? LAST-YEAR))`

(b) `(and (even? LAST-YEAR) (even? THIS-YEAR))`

(c)
```
(cond
    ((number? "Twenty Eleven") "Groucho")
    ((number? "2011") "Harpo")
    ((number? THIS-YEAR) "Zeppo")
    (else "Chico"))
```

(d) `(not (> THIS-YEAR LAST-YEAR))`

(e)
```
(+ 10
    (cond
       ((= THIS-YEAR (+ 1 LAST-YEAR)) THIS-YEAR)
       ((< LAST-YEAR (+ 1 THIS-YEAR)) LAST-YEAR)
       (else 2000)))
```

Problem 2 (4 points)

Each of the following statements relates to one of the video lectures (why study programming, why scheme, or functional programming). Mark each statement as accurate or inaccurate, and for the inaccurate ones, please say in a few words what's wrong with the statement, according to the video.

(a) People who don't plan careers as software developers or software engineers shouldn't bother learning computer programming.

(b) Scheme, a programming language in the Lisp family, is the best programming language to use for nearly any programming task.

(c) Functional programs are evaluated by substituting equals for equals; we can be more confident they're correct than with conventional (imperative) programs, whose memory can be changed by any part of the program.

(d) We used to make computers faster by making them smaller; shortening the distance that a signal travels lets it arrive sooner. But we can't get much smaller, so these days, we have multiple processors (or "cores").

(e) Elephant gestation is an example of a potentially parallelizable process; it goes faster with more than one processor.

Problem 3 (12 points)

On Anteater Airlines, a passenger earns frequent flyer points depending on the length of the trip; the number of points earned per mile depends on the class of service (first, business, economy, or non-refundable).

(a) (3 points) In the function below, fill in the blanks (one string, number, or identifier name per blank) according to the contract and purpose statement shown.

```
; points-per-mile:  string(class of service)  ->  number
; If input is "first", output is 5; if input is "business", output is 3;
; if input is "economy", output is 1; if input is "non-refundable", output is 0.5.
```

```
(define points-per-mile
   (lambda (class)
      (cond
         ((string=? _____________ _____________) _____________)
         ((string=? _____________ _____________) _____________)
         ((string=? _____________ _____________) _____________)
         ((string=? _____________ _____________) _____________))))
```

(b) (9 points) The contract and purpose statement below describe a function that will compute the number of frequent flyer points a passenger earns on a flight.

```
; ff-points:  number(miles)  string(class)  ->  number
; Return the number of frequent flyer points earned for the given number of miles
; flown in the given class.
```

(b.1) (4 points) Write examples/tests for `ff-points` in the form of `check-expect` expressions. Your examples/tests should provide complete coverage of the code (i.e., every part of the code for this problem should be evaluated at least once, so there would be no black highlighting in DrRacket).

(b.2) (5 points) Write the definition of `ff-points` according to the contract and purpose given above. Use the function from part **(a)** if and when it's appropriate.